METHOD AND PROGRAM PRODUCT FOR COSTING AND PLANNING THE RE-HOSTING OF COMPUTER-BASED APPLICATIONS

REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of provisional application Serial No. 60/457,155, filed March 24, 2003.

BACKGROUND OF THE INVENTION

1. Technical Field of the Invention

[0002] The invention relates generally to the field of methods for re-hosting computer-based applications. More specifically, the invention relates to a method and program product for costing and planning the re-hosting, or migration, of computer-based applications from source computer-based platforms to target computer-based platforms.

2. Description of Related Art

[0003] The use of computer systems and applications greatly simplifies the storage and processing of data. Nevertheless, the large volume of data that a business processes in a short time period can cause the business to become entrenched in its use of a particular platform or computer-based environment. By the time a significant advance is made, the business may be so mired within an antiquated system that it is forced to lose money each year due to the inefficiencies of the system. In response to this problem, methods and systems for migrating applications among computer-based platforms have been developed.

[0004] Re-hosting applications from a source platform to a target platform can be expensive, but a time typically arrives when the cost of continuing inefficiencies outweighs the cost of migration. A business must be able to determine when it reaches this condition, in order to obtain the most benefit from its time and monetary expenditures on a migration. Nevertheless, while methods for migration have grown, businesses have been unable to estimate the cost for migration of applications within useful tolerances. Hence, businesses are prone to waste their resources by significantly undershooting or overshooting the point when migration becomes a cost-effective solution.

[0005] In addition to the cost of a migration project, a business also needs to allow for downtime and other lapses in its application and computer system operations. Again, current migration methods overlook this logistical question and only focus on the end result. A business undergoing migration can suffer needlessly, then, due to an inability to plan precisely for the length of interruption in its operations. This adds to the true cost of migration, forcing some businesses to undergo migration of applications either sooner, or later, than it becomes cost-effective.

[0006] As a result, there exists a great need in the art for a method of estimating the cost of application migration, on a per application basis, within specified tolerances. Additionally, there is a great need in the art for a method of estimating the time required for application migration, on a per application basis, within specified tolerances. The method must provide a thorough analysis of factors that affect the cost and time required for the migration of individual applications, in order to overcome the inconsistencies and inaccuracies of *ad hoc* plans and cost estimates.

SUMMARY OF THE INVENTION

[0007] The current invention is directed to a method of estimating the cost of application migration, on a per application basis. The invention is also directed toward a method for estimating the time required for application migration, on a per application basis. The method provides allows varying degrees of thoroughness in its analysis of migration attributes that affect the cost and time required for the migration of individual applications between platforms. In this way, the invention produces consistent and reliable results with a degree of accuracy that is genuinely valuable to the particular user.

[0008] The current invention provides a computer-implemented method for estimating the cost and/or time requirements for migrating a computer-based application from a source platform to a target platform. The invented method comprises the steps of receiving identifications of said migration tasks; correlating base costs to respective ones of said migration tasks; receiving identifications of migration attributes; correlating cost factors to respective ones of said migration tasks, each cost factor indicating an amount by which a migration attribute changes the base cost of a migration task; and estimating a cost for each migration task, by applying all cost factors for the migration task to the base cost of the migration task.

[0009] The migration attributes that may be considered in estimating cost may comprise one or more attributes, such as hardware attributes, operating system attributes, application attributes, environment attributes, source code attributes, complexity attributes and testing attributes, as further described herein. One or more of the base costs and cost factors may be received from a user.

[0010] The invented method may also comprise applying tolerances to one or more of the estimated costs and total cost, such that one or more of these is returned as a cost range. At least one assessment type may be chosen or defined by a user, such that the assessment observes the user's desired degree of accuracy for the total cost and the cost of each migration task.

[0011] The invented method may comprise, in addition or in the alternative to cost estimation, steps for estimating time requirements for a migration. The steps required for estimating time may comprise correlating base time requirements to respective ones of said migration tasks; correlating time factors to respective ones of said migration tasks, each time factor indicating an amount by which a migration attribute changes the base time requirement for a migration task; and estimating a time requirement for each identified migration task, by applying all time factors assigned to the migration task to the base time requirement of the migration task.

[0012] The migration attributes that may be considered in estimating time requirements may comprise one or more attributes, such as hardware attributes, operating system attributes, application attributes, environment attributes, source code attributes, complexity attributes and testing attributes, as further described herein. One or more of the base time requirements and time factors may be received from a user.

[0013] The invented method may also comprise applying tolerances to one or more of the estimated time requirements and total time requirement, such that one or more of these is returned as a time range. At least one assessment type may be chosen or defined by a user, such that the assessment observes the user's desired degree of accuracy for the total time requirement and the time requirement for each migration task.

[0014] Where both time and cost for each migration task and/or for a total migration are estimated, they may be output on a common assessment.

[0015] Individual aspects or functions of the invented method may be embodied in computer-readable program products. Hence the current invention is also directed to computer-readable program code means for implementing and executing the steps of the methods disclosed, in a manner that will be readily known to those skilled in the art.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] FIG. 1 is a flow diagram illustrating the invented method, wherein a cost and/or time estimate for the migration of an application from a source platform to a target platform is output.

[0017] FIG. 2 is an illustration of an example embodiment of an assessment template.

[0018] FIG. 3 is a flow diagram further illustrating steps of the invented method, wherein a Type C assessment is output, in accordance with the current invention.

[0019] FIG. 4A is an illustration of an example embodiment of a base cost matrix.

[0020] FIG. 4B is an illustration of an example embodiment of a complexity cost factor matrix.

[0021] FIG. 5 is an illustration of an example embodiment of a hardware cost factor matrix.

[0022] FIG. 6 is an illustration of an example embodiment of an OS cost factor matrix.

[0023] FIG. 7 is an illustration of an example embodiment of an application cost factor matrix.

[0024] FIG. 8 is an illustration of an example embodiment of an environment cost factor matrix.

[0025] FIG. 9A is an illustration of an example embodiment of a base time requirement matrix.

[0026] FIG. 9B is an illustration of an example embodiment of a complexity time factor matrix.

[0027] FIG. 10 is an illustration of an example embodiment of a hardware time factor matrix.

[0028] FIG. 11 is an illustration of an example embodiment of an OS time factor matrix.

[0029] FIG. 12 is an illustration of an example embodiment of an application time factor matrix.

[0030] FIG. 13 is an illustration of an example embodiment of an environment time factor matrix.

[0031] FIG. 14 is a flow diagram illustrating steps of the invented method, wherein a Type B assessment is output, in accordance with the current invention.

[0032] FIG. 15 is an illustration of an example embodiment of a code cost factor matrix.

[0033] FIG. 16 is a flow diagram illustrating steps of the invented method, wherein a Type A assessment is output, in accordance with the current invention.

[0034] FIG. 17 is an illustration of an example embodiment of a code time factor matrix.

[0035] FIG. 18 is an illustration of an example embodiment of a testing cost factor matrix.

[0036] FIG. 19 is an illustration of an example embodiment of a testing time factor matrix.

DETAILED DESCRIPTION OF THE INVENTION

[0037] Referring now to the drawings, the invention is directed to a method for costing and planning the migration of an application among computer-based environments. The individual steps of the method are preferably executed by at least one computer software program embodied on a computer-readable medium, without regard to the operating system of the computer or the language of the software programs. It will be appreciated by those skilled in the art that some steps of the method may be performed in series, and some in parallel or in series. Additionally, the order of the steps may in some instances be changed, without departing from the scope or advantages of the current invention. The order in the following embodiments is given only for ease of explanation.

[0038] FIG. 1 is a flow diagram illustrating a method for costing and planning the migration of an application from a source computer-based environment to a target computer-based environment, in accordance with the current invention. In accordance with step **101**, a set of base variables is received. The base variables comprise an application to be migrated, a source platform on which the application currently operates, and a target platform for the application.

[0039] In accordance with step **102**, an assessment type is received. The assessment types are referred to herein as Type C, Type B, and Type A assessments. An assessment may comprise a cost estimate, a time estimate, or both cost and time estimates, for migrating the application from the source platform to the target platform. Assessment types may be delineated by the type or amount of information processed, in order to calculate the cost and/or time estimate returned by the assessment. Assessment types may also be delineated by the degree of accuracy in cost and/or time that the assessment is intended to entail. For purposes of this description, assessments are delineated by their intended accuracy, and, hence, by the type and amount of information gathered.

[0040] In accordance with step **103**, an identifier for at least one migration task is received. The migration tasks may be user-defined, or they may be chosen from a list of pre-defined migration tasks. Alternatively, a set of migration tasks may be associated with each assessment type, such that they are received automatically upon receipt of an assessment type. The individual migration tasks involved in migrating a specified application from a particular source platform to a particular target platform are readily known to those skilled in the art. Such migration tasks may comprise, for example, system building, project management, ramp up, baseline testing, migration of the application from the source platform to the target platform, system testing after migration, delivery of the migrated application, acceptance testing to validate that the application can process data, project completion and sign-off, exporting data from the source platform and importing the data to the application on the target platform, redirecting user terminals to the target platform, replacing third party products that cannot be ported to the target platform, and any other individual migration tasks that may be involved in the migration of a computer-based application from a source platform to a target platform.

[0041] In accordance with step **104**, at least one assessment template may be created. An assessment template contains the names of the individual migration tasks that will be involved in the migration, cross-referenced to the costs of such migration tasks. The migration tasks may also be cross-referenced to time requirements for each migration task, such as duration of each task in work hours or days, and start and finish times. The migration tasks may be cross-referenced to any other information suitable for the planning and costing of individual migration project tasks. In one embodiment, the assessment template comprises a form, such as that shown by FIG. 2. Those skilled in the art will appreciate that this embodiment of the template is illustrative, and that other templates, including relational and associative forms may be used without departing from the scope of the invented method.

[0042] The assessment template may be created upon receiving a plurality of migration tasks. Alternatively, an assessment template may be chosen from a plurality of pre-defined templates, each containing a different subset of migration tasks.

[0043] The cost of each migration task in the assessment template is shown in a column separate from that of the migration tasks, such that each cost is represented in the same row of the template as the migration task to which it corresponds. For purposes of this description, the cost of each migration task 'i' will be represented by the function $f(c_i)$. The variable 'i' may actually be represented as a priority number or any other suitable number for sequentially identifying the migration tasks represented in the assessment template, such that the total cost of the migration process can be achieved by summing the costs of migration tasks, 1...i.

[0044] The template may also include a breakdown of costs, such as labor and materials, with or without reference to specific tasks.

[0045] At least one time requirement for each migration task in the assessment template may be shown in a column separate from those of the migration tasks and costs, such that a time requirement is represented in the same row of the assessment template as the migration task to which it corresponds. For purposes of this description, the time requirement for each individual task will be represented by the function $f(t_i)$. The variable 'i' may actually be represented as a priority number or any other suitable number for sequentially identifying the migration tasks represented in the assessment template, such that the total time for the migration process can be achieved by summing the time requirements for migration tasks, 1...i. Where both time and cost are estimated on the template, the variable 'i' shall be identical for both $f(t_i)$ and $f(c_i)$.

[0046] In accordance with step **105**, a base cost is assigned to each migration task. Where time is being estimated, a base time is assigned to each migration task. The base cost comprises the average base cost for the migration task, when migrating the application received in step **101** between the source and target platforms received in step **101**. The base time comprises the average base time requirement for the migration task, when migrating the application received in step **101** between the source and target platforms received in step **101**. Base costs and base times may be user-defined; may be chosen from a database; or may be automatically assigned from a database upon receipt of the base variables. Base costs and base times may vary depending upon the type of assessment to be returned.

[0047] In accordance with step **106**, attributes relevant to the migration are received. Migration attributes may include hardware attributes, operating system attributes, application attributes, environment attributes, source code attributes, testing attributes, or any combination of these. The various migration attributes are described in further detail below.

[0048] For each migration attribute received, a cost factor is assigned to each identified migration task, in accordance with step **107**. Each cost factor represents the amount by which the attribute changes the base cost of a migration task. Where time is being estimated, a time factor is assigned to each migration task. Each time factor represents the amount by which the attribute changes the base time of each migration task. Cost and time factors may comprise proportions that are greater than, less than, or equal to, one. Cost and time factors may be input by users; they may be chosen from a database of cost and time factors; or they may be automatically called from a database in the forms of cost and time factor matrices relevant to the attributes and base variables, as further described below.

[0049] In accordance with step **108**, the cost factors assigned to each migration task are applied to the base cost for the migration task, as further described herein. This results in an estimated cost for each migration task. Time factors assigned to each migration task are applied to the base times for the migration task, as further described herein. In accordance with step **109**, tolerances may be applied to the cost and/or time for individual migration tasks. The cost and/or time for the migration tasks are then summed to achieve a total cost and/or time requirement for the migration. Finally, in accordance with step **110**, an assessment of the type received in step **102** is output, containing the estimated cost and/or time requirement for the migration. The assessment may conform to the assessment template created in step **104**, if any. The estimates for cost and time may comprise fixed numbers or ranges. The assessment may also contain cost and/or time estimates for individual migration tasks, which estimates may be fixed numbers, ranges, or a combination of fixed numbers and ranges.

[0050] The steps of the invented method are described in further detail below, with specific references to the different types of assessments and the assignment of base costs and times, and cost and time factors.

[0051] FIG. 3 is a flow diagram illustrating a method for returning a Type C assessment for the migration of an application from a source platform to a target platform, in accordance with the current invention. In accordance with step **301**, base variables are received, which comprise an application to be migrated, a source platform on which the application currently operates, and a target platform on which the application is to be re-hosted.

[0052] The application variable may comprise a specific application name and version. Alternatively, the application may include the function of the application and (if more than one function) its component parts, such as production control, batch processing, data mining, online transaction processing (OLTP), or other functions. In the preferred embodiment of the invention, the application variable comprises the name, version and function(s) of the application and its component parts.

[0053] The source and target platform variables may comprise any platforms suitable for operating the computer-based application to be migrated. These platforms may include, for example, UNIX (generic or variants), OpenVMS, IBM AS/400 or iSeries, HP 300 MPE, IBM Mainframe, and other platforms that are readily known to those skilled in the art.

[0054] In accordance with step **302**, a user's choice of a Type C assessment is received. In accordance with step **303**, an identifier for at least one migration task is received, as described with reference to FIG. 1. In accordance with step **304**, an assessment template may be created in the manner described with reference to FIG. 1. The assessment template for a Type C assessment preferably comprises a high-level representation of migration tasks, as distinguished from a Type B or Type A assessment template. A Type C template may also comprise just a total figure for cost and/or time.

[0055] In accordance with step **305**, base costs and/or times are assigned to each migration task. For a Type C assessment, the base variables are preferably compared with at least one base cost matrix in a database. Each base cost matrix sets forth an average base cost for migration tasks, such as those listed previously. A base cost matrix may conform substantially to that shown in FIG. 4A, wherein migration tasks are listed in one column, and a base cost is shown in the row containing a migration task to which the base cost corresponds. A base cost , $^{i}\beta$, is extracted from the database, for each migration task, 'i', received in step **303**. Note that each base cost matrix may contain base cost data for more migration tasks than are received in step **303**.

[0056] Each base cost matrix may set forth average costs for migration tasks, with respect to specific applications. Alternatively, each base cost matrix may set forth average base costs for migration tasks, with respect to the degree of commonality of various applications. For example, base cost matrices may be built for standard applications (for example, word processors and data storage), non-standard (such as compilers and communication software), and custom-made applications. Matrices may be built to reflect additional qualitative categories of applications that have a quantifiably distinguishable effect on the cost of at least one migration task.

[0057] Where time requirements are being estimated, base time requirements are assigned to each migration task. For a Type C assessment, the base variables are preferably compared with at least one base time matrix in a database. Each base cost matrix sets forth an average base cost for migration tasks, such as those listed previously. A base time requirement matrix may conform substantially to that shown in FIG. 9A, wherein migration tasks are listed in one column, and a base time requirement is shown in the row containing a migration task to which the base time requirement corresponds.

A base time, ${}^{i}T$, is extracted from the database, for each migration task, 'i', received in step **303**. Note that each base cost matrix may contain base time data for more migration tasks than are received in step **303**.

[0058] Each base time requirement matrix may set forth average time requirements for migration tasks, with respect to specific applications. Alternatively, each base time matrix may set forth average base time requirements for migration tasks, with respect to the degree of commonality of various applications. For example, base time matrices may be built for standard applications (for example, word processors and data storage), non-standard (such as compilers and communication software), and custom-made applications. Matrices may be built to reflect additional qualitative categories of applications that have a quantifiably distinguishable effect on the time requirement for at least one migration task.

[0059] In accordance with step **306**, at least one migration attribute is received. The migration attributes may comprise, for example, hardware attributes, current operating system attributes, attributes of the application to be migrated, environment attributes, or testing attributes. Hardware attributes include elements of source and target platform hardware architecture, which affect the complexity of migrating an application or porting data between the source and target platforms. Examples of hardware attributes may include, for example, clustering, disks, external interfaces, archive media, memory, networking, media, processors, terminal types (e.g., dumb, emulated), workstations, third-party hardware elements, byte storage sequence (e.g., big endian, little endian) and other aspects of hardware architecture that impact the process of migrating an application or porting data a source platform and a target platform.

[0060] Operating system attributes may comprise the type and version of the source platform's current operating system(s).

[0061] The application attributes comprise attributes of the application that may affect the cost and plan for migrating the application, other than the overall function(s) of the application. Application attributes may comprise, for example, whether the application is real-time; mission critical; user interactive; used for production control, batch processing, data mining, on-line transaction processing (OLTP), or other relevant attributes that may affect the cost and plan for migrating the application. Application attributes may be expressed in descriptive form, or as yes-no (or other analogous binary system) indicators for pre-defined attributes.

[0062] Environment attributes may comprise any attributes of the software environment, in which the application is operated on the source platform, that may affect the cost or plan for migrating the application. Environment attributes may comprise, for example, development languages; databases and other data storage and access attributes; user interfaces; communication transport protocols; networking attributes; shell script usage; security attributes; batch processing characteristics; and other attributes of the software environment in which the application is operated on the source platform, that may affect the cost or plan for migrating the application.

[0063] Testing can be used as a measurement of progress during migration, and it can limit debugging time afterward. Testing itself, however, can have a significant impact upon the time and cost for a migration. Testing during a migration may comprise baseline testing, i.e., testing the application on the source platform before it is migrated. Testing may also comprise system testing, i.e., testing the application after migration. A set of criteria may be established for verifying proper performance of the application. Both the baseline testing results and the system testing results are compared with the criteria, such that proper performance of the application can be verified both before and after the migration of the application between platforms.

[0064] Testing attributes may comprise any inventory and criteria that might impact upon the length of time or cost for baseline or system testing. Such information may include, for example, the need to create testing programs; whether baseline or system testing requires visits to other sites; the need to setup the application or the testing programs prior to baseline testing; any need to rebuild the application in a clean environment prior to testing; the testing process itself; verifying test results and comparing them to criteria; and debugging of the application after migration. Testing attributes may also comprise the types of testing to be performed, such as unit tests (subroutine level) and whether unit testing will be performed as the application is ported to the new platform; functional tests (tests of user functions) testing of batch processes; integration tests (program level tests of the application's external interfaces); regression tests (movement of data between user functions); performance tests; system management tests (start-up, shutdown, etc.); and hardware tests.

[0065] In accordance with step **307**, cost and/or time factors corresponding to each migration attribute received in step **306**, and corresponding to the base variables, are assigned to each migration task. For a Type C assessment, the assignment of cost factors is preferably achieved by a user choosing attributes from pre-defined lists. The attributes are then compared with at least one cost factor matrix in a database. Each cost factor matrix contains cost factors for various migration tasks, when at least one attribute is relevant to migrating a specified application from a specified source platform to a specified target platform. Each cost factor matrix may set forth factors for migration tasks, with respect to specific applications. Alternatively, each cost factor matrix may set forth cost factors for migration tasks, with respect to the degree of commonality of various applications. For example, cost factor matrices may be built for standard applications (for example, word processors and data storage), non-standard (such as compilers and communication software), and custom-made applications. Matrices may be built to reflect additional qualitative categories of applications that have a quantifiably distinguishable effect on the cost of at least one migration task. Various cost factor matrices are described in further detail, below.

[0066] One example of a cost factor matrix contemplated for returning a Type C assessment is a complexity cost factor matrix. Each complexity cost factor matrix sets forth factors, comprising the amounts by which the costs of various migration tasks are changed by the complexity of migrating an application between a source platform and a target platform. A complexity cost factor matrix may conform substantially to that shown in FIG. 4B, wherein a complexity cost factor, ${}^{i}\lambda_{S\text{-}T}$, is shown at the intersection of each row containing a migration task, 'i', and a column containing a source-target combination, 'S-T'. Note that a complexity cost factor matrix may contain complexity cost factors for more migration tasks than are received in step **303**.

[0067] The factors may comprise proportions by which the base costs of migration tasks may be multiplied to reflect the effect of the source-target combination in increasing, decreasing or not affecting, the average base cost of migration tasks during a migration process. The effect of various source-target combinations upon the base cost of each migration task, and hence the development of each complexity cost factor, $^{i}\lambda_{S\text{-}T}$, will be readily appreciated by those skilled in the art.

[0068] A single universal complexity cost factor matrix may be used, or many complexity cost factor matrices may be used, each one containing factors for only one type of source-target combination, such as UNIX-UNIX; Generic UNIX-UNIX Variant; AIXv(x)-AIXv(x+1); Windows XP-Solaris/Intel; and the like. Species of source and target platforms may be particularized (such as Bull GCOS, DG DG-UX, FreeBSD, HP HP-UX, HP MPE, HP NSK, HP OpenVMS/Alpha, HP OpenVMS/VAX, HP Tru64 UNIX, HP VMS/VAX, IBM AIX, IBM DOS/VSE, IBM DYNIX/ptx, IBM MVS, IBM OS/390, IBM zOS, Intel Linux, SCO UnixWare, SGI Irix, Solaris/Intel, Solaris/SPARC, Windows 9x, Windows NT/200, Windows XP and the like). Species of source and target platforms may be non-particularized (such as common platform, uncommon platform, custom platform, proprietary platform, and the like). Particularized and non-particularized species may be combined on a single complexity cost factor matrix and in individual source-target combinations (such as WinXP-proprietary platform).

[0069] Once at least one complexity cost factor matrix is identified that reflects the source-target combination received in step **301**, a complexity cost factor, ${}^{i}\lambda_{S\text{-}T}$, for each migration task, 'i', in the assessment template is obtained from the complexity cost factor matrix or matrices. Note that ${}^{i}\lambda_{S\text{-}T}$ may differ for each migration task.

[0070] Another type of cost factor matrix that may be used in returning a Type C assessment is a hardware cost factor matrix. Each hardware cost factor matrix sets forth factors, comprising the amounts by which the costs of various migration tasks are changed due to the presence of at least one hardware attribute during migration. A hardware cost factor matrix may conform substantially to that shown in FIG. 5, wherein a hardware cost factor, ${}^{i}\hbar_{k}$, is shown at the intersection of each row containing a migration task, 'i', and a column containing a hardware attribute combination, 'k'. Note that a hardware cost factor matrix may contain hardware cost factors for more migration tasks than are received in step **303**.

[0071] The factors may comprise proportions by which the base costs of migration tasks may be multiplied to reflect the effect of at least one hardware attribute in increasing, decreasing or not affecting, the average base costs of the migration tasks during a migration process. The effect of various hardware attribute combinations upon the base cost of each migration task, and hence the development of each hardware cost factor, ${}^{i}\hbar_{k}$, will be readily appreciated by those skilled in the art.

[0072] A single universal hardware cost factor matrix may be used, or many hardware cost factor matrices may be used, each one containing factors for only one type of hardware attribute combination, such as external interface combinations, media combinations, byte storage sequencing combinations, and the like. Species of hardware attributes may be particularized (such as big endian to little endian, and the like). Species of hardware attributes may be non-particularized (such as, standard connectivity to non-standard connectivity, common processors to custom processors, and the like). Particularized and non-particularized species may be combined on a single hardware cost factor matrix.

[0073] Once at least one hardware cost factor matrix is identified containing the hardware attribute combinations, 1...k, which were received in accordance with step **306**, the hardware cost factors, $^{i}\hbar_{1}$ ... $^{i}\hbar_{k}$, for each migration task in the assessment template are obtained from the hardware cost factor matrix or matrices. Note that $^{i}\hbar_{1}$ ... $^{i}\hbar_{k}$ may differ for each migration task.

[0074] Another type of cost factor matrix that may be used in returning a Type C assessment is an operating system (OS) cost factor matrix. Each OS cost factor matrix sets forth factors, comprising the amounts by which the costs of various migration tasks are changed due to the operating system operated on the source platform (due to, for example, threads, embedded SQL/RDBMS access, kernel mode routines, clustering functionality, operating system intrinsics, library functions, etc.). An OS cost factor matrix may conform substantially to that shown in FIG. 6, wherein an OS cost factor, $^{i}\omega_{m}$, is shown at the intersection of each row containing a migration task, 'i', and a column containing an operating system, 'm'. Note that an OS cost factor matrix may contain OS cost factors for more migration tasks than are received in step **303**.

[0075] The factors may comprise proportions by which the base costs of migration tasks may be multiplied to reflect the effect of the OS operated on the source platform in increasing, decreasing or not affecting, the average base costs of the migration tasks during a migration process. The effect of various operating systems upon the base cost of each migration task, and hence the development of each OS cost factor, ${}^{i}\omega_{m}$, will be readily appreciated by those skilled in the art.

[0076] A single universal OS cost factor matrix may be used, or many OS cost factor matrices may be used, each one containing factors for only one type of operating system. Species of operating systems may be particularized (such as Microsoft Windows® NT, Sun Solaris® and the like). Species of operating systems may be non-particularized (such as, standard, non-standard, custom, and the like). Particularized and non-particularized species may be combined on a single OS cost factor matrix.

[0077] Once at least one OS cost factor matrix is identified containing the operating systems, 1...m, which were received in accordance with step **306**, the OS cost factors, ${}^{i}\omega_{1} \ldots {}^{i}\omega_{m}$, for each migration task in the assessment template are obtained from the OS cost factor matrix or matrices. Note that ${}^{i}\omega_{1} \ldots {}^{i}\omega_{m}$, may differ for each migration task.

[0078] Another type of cost factor matrix that may be used in returning a Type C assessment is an application cost factor matrix. Each application cost factor matrix sets forth factors, comprising the amounts by which the costs of various migration tasks are changed due to at least one attribute of the application being migrated. An application cost factor matrix may conform substantially to that shown in FIG. 7, wherein an application cost factor, ${}^{i}\alpha_n$, is shown at the intersection of each row containing a migration task, 'i', and a column containing an application attribute, 'n'. Note that an application cost factor matrix may contain application cost factors for more migration tasks than are received in step **303**.

[0079] The factors may comprise proportions by which the base costs of migration tasks may be multiplied to reflect the effect of application attributes in increasing, decreasing, or not affecting, the average base costs of the migration tasks during a migration process. The effect of each application attribute upon the base cost of each migration task, and hence the development of each application cost factor, ${}^{i}\alpha_n$, will be readily appreciated by those skilled in the art.

[0080] A single universal application cost factor matrix may be used, or many application cost factor matrices may be used, each one for a single application attribute, such as real-time operation, OLTP, and the like. Species of applications may be non-particularized (such as, standard, non-standard, custom, and the like). Particularized and non-particularized species may be combined on a single application cost factor matrix.

[0081] Once at least one application cost factor matrix is identified containing the application attributes, 1…n, which were received in accordance with step **306**, the application cost factors, ${}^{i}\alpha_1 \ldots {}^{i}\alpha_n$, for each migration task in the assessment template are obtained from the application cost factor matrix or matrices. Note that ${}^{i}\alpha_1 \ldots {}^{i}\alpha_n$, may differ for each migration task.

[0082] Another type of cost factor matrix that may be used in returning a Type C assessment is an environment cost factor matrix. Each environment cost factor matrix sets forth factors, comprising the amounts by which the costs of various migration tasks are changed due to at least one environment attribute. An environment cost factor matrix may conform substantially to that shown in FIG. 8, wherein an environment cost factor, ${}^{i}\epsilon_{p}$, is shown at the intersection of each row containing a migration task, ‘i’, and a column containing an environment attribute, ‘p’. Note that an environment cost factor matrix may contain environment cost factors for more migration tasks than are received in step **303**.

[0083] The factors may comprise proportions by which the base costs of migration tasks may be multiplied to reflect the effect of environment attributes in increasing, decreasing, or not affecting, the average base costs of the migration tasks during a migration process. The effect of each environment attribute upon the base cost of each migration task, and hence the development of each environment cost factor, ${}^{i}\epsilon_{p}$, will be readily appreciated by those skilled in the art.

[0084] A single universal environment cost factor matrix may be used, or many environment cost factor matrices may be used, each one containing factors for only one type of environment attribute, such as external interface, media, and the like. Species of environment attributes may be particularized (such as 128-bit SSL security, hypertext transfer protocol usage, COBOL language and the like). Species of environment attributes may be non-particularized (such as, standard languages, non-standard security, custom communication transfer protocols, and the like). Particularized and non-particularized species may be combined on a single environment cost factor matrix.

[0085] Once at least one environment cost factor matrix is identified containing the environment attributes, 1...p, which were received in accordance with step **306**, the environment cost factors, ${}^{i}\epsilon_1 \ldots {}^{i}\epsilon_p$, for each migration task in the assessment template are obtained from the environment cost factor matrix or matrices. Note that ${}^{i}\epsilon_1 \ldots {}^{i}\epsilon_p$, may differ for each migration task.

[0086] Another type of cost factor matrix that may be used in returning a Type C assessment is a testing cost factor matrix. Each testing cost factor matrix sets forth factors, comprising the amounts by which the costs of various migration tasks are changed due to at least one testing attribute. A testing cost factor matrix may conform substantially to that shown in FIG. 18, wherein a testing cost factor, ${}^{i}\eta_r$, is shown at the intersection of each row containing a migration task, 'i', and a column containing an environment attribute, 'r'. Note that the effects of testing attributes may only affect, for example, the baseline testing and system testing costs. This example is shown in FIG. 18, where $i = 2$ or 4, because these are the task numbers given to baseline and system testing in the example assessment template shown in FIG. 2.

[0087] The factors may comprise proportions by which the base cost of baseline test and/or system test may be multiplied to reflect the effect of the testing attribute in increasing, decreasing, or not affecting, the average base cost of the baseline testing and/or system testing during a migration process. The effect of each testing attribute upon the base cost of baseline testing and system testing, and hence the development of each environment cost factor, ${}^{i}\eta_r$, will be readily appreciated by those skilled in the art.

[0088] A single universal testing cost factor matrix may be used, or many environment cost factor matrices may be used, each one containing factors for only one type of testing attribute, such as re-building the application, use of unit testing, creation of testing programs, testing sites, testing criteria and the like. Species of testing attributes may be particularized (such as no re-building, progressive unit testing, perform baseline test off-site and the like). Species of testing attributes may be non-particularized (such as, standard testing suites, non-standard performance criteria, custom hardware simulation during testing, and the like). Particularized and non-particularized species may be combined on a single testing cost factor matrix.

[0089] Once at least one testing cost factor matrix is identified containing the testing attributes, 1…r, which were received in accordance with step **306**, the testing cost factors, ${}^{i}\eta_1 \ldots {}^{i}\eta_r$, for baseline and system testing are obtained from the testing cost factor matrix or matrices. Note that ${}^{i}\eta_1 \ldots {}^{i}\eta_r$, may differ for each type of testing.

[0090] In accordance with step **308**, the cost for the migration is estimated. The calculation of estimated cost is achieved by applying the cost factors assigned in step **307** to the base costs assigned in step **305**. The cost $f(c_i)$ for each individual migration task received in step **303**, other than baseline test and system test, is estimated as follows:

$$f(c_i) = {}^{i}\beta\, {}^{i}\lambda_{S\text{-}T}\, ({}^{i}\hbar_1\, {}^{i}\hbar_2 \ldots {}^{i}\hbar_n)({}^{i}\omega_1\, {}^{i}\omega_2 \ldots {}^{i}\omega_m)({}^{i}\alpha_1\, {}^{i}\alpha_2 \ldots {}^{i}\alpha_n)({}^{i}\epsilon_1\, {}^{i}\epsilon_2 \ldots {}^{i}\epsilon_p)$$

[0091] The cost $f(c_2)$ for baseline test is estimated as follows:

$$f(c_2) = {}^{2}\beta\, {}^{2}\lambda_{S\text{-}T}\, ({}^{2}\hbar_1\, {}^{2}\hbar_2 \ldots {}^{2}\hbar_n)({}^{2}\omega_1\, {}^{2}\omega_2 \ldots {}^{2}\omega_m)({}^{2}\alpha_1\, {}^{2}\alpha_2 \ldots {}^{2}\alpha_n)({}^{2}\epsilon_1\, {}^{2}\epsilon_2 \ldots {}^{2}\epsilon_p)({}^{2}\eta_1\, {}^{2}\eta_2 \ldots {}^{2}\eta_r).$$

[0092] The cost $f(c_2)$ for system test is estimated as follows:

$$f(c_4) = {}^4\beta\, {}^4\lambda_{S\text{-}T}\, ({}^4\hbar_1\, {}^4\hbar_2 \ldots {}^4\hbar_n)({}^4\omega_1\, {}^4\omega_2 \ldots {}^4\omega_m)({}^4\alpha_1\, {}^4\alpha_2 \ldots {}^4\alpha_n)({}^4\epsilon_1\, {}^4\epsilon_2 \ldots {}^4\epsilon_p)({}^4\eta_1\, {}^4\eta_2 \ldots\, {}^4\eta_r).$$

[0093] The cost estimate for the migration process shown on the assessment template is then estimated as follows:

$$\sum_{i=1}^{i} f(c_i)$$

[0094] Where time requirements are estimated, step **307** includes assignment of time factors to each migration task. For a Type C assessment, the assignment of time factors is preferably achieved by a user choosing attributes from pre-defined lists. The attributes are then compared with at least one time factor matrix in a database. Each time factor matrix contains time factors for various migration tasks, when at least one attribute is relevant to migrating a specified application from a specified source platform to a specified target platform. Each time factor matrix may set forth factors for migration tasks, with respect to specific applications. Alternatively, each time factor matrix may set forth time factors for migration tasks, with respect to the degree of commonality of various applications. For example, time factor matrices may be built for standard applications (for example, word processors and data storage), non-standard (such as compilers and communication software), and custom-made applications. Matrices may be built to reflect additional qualitative categories of applications that have a quantifiably distinguishable effect on the time requirement for at least one migration task.

[0095] Various time factor matrices are described in further detail, below.

One example of a time factor matrix contemplated for returning a Type C assessment is a complexity time factor matrix. Each complexity time factor matrix sets forth factors, comprising the amounts by which the time requirements for various migration tasks are changed by the complexity of migrating an application between a source platform and a target platform. A complexity time factor matrix may conform substantially to that shown in FIG. 9B, wherein a complexity time factor, ${}^{i}\psi_{S\text{-}T}$, is shown at the intersection of each row containing a migration task, 'i', and a column containing a source-target combination, 'S-T'. Note that a complexity time factor matrix may contain complexity cost factors for more migration tasks than are received in step **303**.

[0096] The complexity time factors may comprise proportions by which the base time requirements for migration tasks may be multiplied to reflect the effect of the source-target combination in increasing, decreasing, or not affecting, the average base time requirement for the migration tasks during a migration process. The effect of each source-target combination upon the base time requirement for each migration task, and hence the development of each complexity time factor, ${}^{i}\psi_{S\text{-}T}$, will be readily appreciated by those skilled in the art.

[0097] A single universal complexity time factor matrix may be used, or many complexity time factor matrices may be used, each one containing factors for only one type of source-target combination, such as UNIX-UNIX; Generic UNIX-UNIX Variant; AIXv(x)-AIXv(x+1); Windows XP-Solaris/Intel; and the like. Species of source and target platforms may be particularized (such as Bull GCOS, DG DG-UX, FreeBSD, HP

HP-UX, HP MPE, HP NSK, HP OpenVMS/Alpha, HP OpenVMS/VAX, HP Tru64 UNIX, HP VMS/VAX, IBM AIX, IBM DOS/VSE, IBM DYNIX/ptx, IBM MVS, IBM OS/390, IBM zOS, Intel Linux, SCO UnixWare, SGI Irix, Solaris/Intel, Solaris/SPARC, Windows 9x, Windows NT/200, Windows XP and the like). Species of source and target platforms may be non-particularized (such as common platform, uncommon platform, custom platform, proprietary platform, and the like). Particularized and non-particularized species may be combined on a single complexity time factor matrix and in individual source-target combinations (such as WinXP-proprietary platform).

[0098] Once at least one complexity time factor matrix is identified that reflects the source-target combination received in step **301**, complexity time factor, ${}^{i}\psi_{S\text{-}T}$, for each migration task in the assessment template is obtained from the complexity time factor matrix or matrices. Note that ${}^{i}\psi_{S\text{-}T}$ may differ for each migration task.

[0099] Another type of time factor matrix that may be used in returning a Type C assessment is a hardware time factor matrix. Each hardware time factor matrix sets forth factors, comprising the amounts by which the time requirements for various migration tasks are changed due to the presence of at least one hardware attribute during migration. A hardware time factor matrix may conform substantially to that shown in FIG. 10, wherein a hardware time factor, ${}^{i}H_{k}$, is shown at the intersection of each row containing a migration task, 'i', and a column containing a hardware attribute combination, 'k'. Note that a hardware time factor matrix may contain hardware time factors for more migration tasks than are received in step **303**.

[0100] The hardware time factors may comprise proportions by which the base time requirements for migration tasks may be multiplied to reflect the effect of the difference in the hardware attributes of the source and target platforms, in increasing, decreasing, or not affecting, the average base time requirement of the migration tasks during a migration process. The effect of each hardware attribute combination upon the base time requirement for each migration task, and hence the development of each hardware time factor, ${}^{i}H_{k}$, will be readily appreciated by those skilled in the art.

[0101] A single universal hardware time factor matrix may be used, or many hardware time factor matrices may be used, each one containing factors for only one type of hardware attribute combination, such as external interface combinations, media combinations, byte storage sequencing combinations, and the like. Species of hardware attributes may be particularized (such as big endian to little endian, and the like). Species of hardware attributes may be non-particularized (such as, standard connectivity to non-standard connectivity, common processors to custom processors, and the like). Particularized and non-particularized species may be combined on a single hardware time factor matrix.

[0102] Once at least one hardware time factor matrix is identified containing the hardware attribute combinations, 1...k, which were received in accordance with step **306**, the hardware time factors, ${}^{i}H_{1}$ ... ${}^{i}H_{k}$, for each migration task in the assessment template are obtained from the hardware time factor matrix or matrices. Note that ${}^{i}H_{1}$ ... ${}^{i}H_{k}$, may differ for each migration task.

[00103] Another type of time factor matrix that may be used in returning a Type C assessment is an operating system (OS) time factor matrix. Each OS time factor matrix sets forth factors, comprising the amounts by which the time requirements for various migration tasks are changed due to the operating system operated on the source platform (due to, for example, threads, embedded SQL/RDBMS access, kernel mode routines, clustering functionality, operating system intrinsics, library functions, etc.). An OS time factor matrix may conform substantially to that shown in FIG. 11, wherein an OS time factor, ${}^{i}\theta_{m}$, is shown at the intersection of each row containing a migration task, 'i', and a column containing an operating system, 'm'. Note that an OS time factor matrix may contain OS time factors for more migration tasks than are received in step **303**.

[0104] The OS time factors may comprise proportions by which the base time requirements for migration tasks may be multiplied to reflect the effect of the dependence of the application upon the initial operating system(s) (due to, for example, threads, embedded SQL/RDBMS access, kernel mode routines, clustering functionality, operating system intrinsics, library functions, etc.) in increasing, decreasing, or not affecting, the average base time requirements for the migration task during a migration process. The effect of each operating system upon the base time requirement for each migration task, and hence the development of each OS time factor, ${}^{i}\theta_{m}$, will be readily appreciated by those skilled in the art.

[0105] A single universal OS time factor matrix may be used, or many OS time factor matrices may be used, each one containing factors for only one type of operating system. Species of operating systems may be particularized (such as Microsoft

Windows® NT, Sun Solaris® and the like). Species of operating systems may be non-particularized (such as, standard, non-standard, custom, and the like). Particularized and non-particularized species may be combined on a single OS time factor matrix.

[0106] Once at least one OS time factor matrix is identified containing the operating systems, 1...m, which were received in accordance with step **306**, the OS time factors, ${}^{i}\theta_1 \ldots {}^{i}\theta_m$, for each migration task in the assessment template are obtained from the OS time factor matrix or matrices. Note that ${}^{i}\theta_1 \ldots {}^{i}\theta_m$, may differ for each migration task.

[0107] Another type of time factor matrix that may be used in returning a Type C assessment is an application time factor matrix. Each application time factor matrix sets forth factors, comprising the amounts by which the time requirements for various migration tasks are changed due to at least one attribute of the application being migrated. An application time factor matrix may conform substantially to that shown in FIG. 12, wherein an application time factor, ${}^{i}д_n$, is shown at the intersection of each row containing a migration task, 'i', and a column containing an application attribute, 'n'. Note that an application time factor matrix may contain application time factors for more migration tasks than are received in step **303**.

[0108] The application time factors may comprise proportions by which the base times of migration tasks may be multiplied to reflect the effect of application attributes in increasing, decreasing, or not affecting, the average base times of the migration tasks during a migration process. The effect of each application attribute upon the base time requirement for each migration task, and hence the development of each application time factor, ${}^{i}Д_n$, will be readily appreciated by those skilled in the art.

[0109] A single universal application time factor matrix may be used, or many application time factor matrices may be used, each one for a single application attribute, such as real-time operation, OLTP, and the like. Species of applications may be non-particularized (such as, standard, non-standard, custom, and the like). Particularized and non-particularized species may be combined on a single application time factor matrix.

[0110] Once at least one application time factor matrix is identified containing the application attributes, 1…n, which were received in accordance with step **306**, the application time factors, ${}^{i}Д_1 \ldots {}^{i}Д_n$, for each migration task in the assessment template are obtained from the application time factor matrix or matrices. Note that ${}^{i}Д_1 \ldots {}^{i}Д_n$ may differ for each migration task.

[0111] Another type of time factor matrix that may be used in returning a Type C assessment is an environment time factor matrix. Each environment time factor matrix sets forth factors, comprising the amounts by which the time requirements for various migration tasks are changed due to at least one environment attribute. An environment time factor matrix may conform substantially to that shown in FIG. 13, wherein a environment time factor, ${}^{i}\chi_p$, is shown at the intersection of each row containing a migration task, 'i', and a column containing an environment attribute, 'p'. Note that an environment time factor matrix may contain environment time factors for more migration tasks than are received in step **303**.

[0112] The factors may comprise proportions by which the base times of migration tasks may be multiplied to reflect the effect of environment attributes in increasing, decreasing, or not affecting, the average base times of the migration tasks during a migration process. The effect of each environment attribute upon the base time requirement for each migration task, and hence the development of each environment time factor, ${}^{i}\chi_{p}$, will be readily appreciated by those skilled in the art.

[0113] A single universal environment time factor matrix may be used, or many environment time factor matrices may be used, each one containing factors for only one type of environment attribute, such as external interface, media, and the like. Species of environment attributes may be particularized (such as 128-bit SSL security, hypertext transfer protocol usage, COBOL language and the like). Species of environment attributes may be non-particularized (such as, standard languages, non-standard security, custom communication transfer protocols, and the like). Particularized and non-particularized species may be combined on a single environment time factor matrix.

[0114] Once at least one environment time factor matrix is identified containing the environment attributes, 1…p, which were received in accordance with step **306**, the environment time factors, ${}^{i}\chi_{1} \ldots {}^{i}\chi_{p}$, for each migration task in the assessment template are obtained from the environment time factor matrix or matrices. Note that ${}^{i}\chi_{1} \ldots {}^{i}\chi_{p}$, may differ for each migration task.

[0115] Another type of time factor matrix that may be used in returning a Type C assessment is a testing time factor matrix. Each testing time factor matrix sets forth factors, comprising the amounts by which the times of various migration tasks are changed due to at least one testing attribute. A testing time factor matrix may conform substantially to that shown in FIG. 19, wherein a testing time factor, ${}^{i}\text{ъ}_{r}$, is shown at the intersection of each row containing a migration task, ‘i’, and a column containing a testing attribute, ‘r’. This example is shown in FIG. 19, where i = 2 or 4, because these are the task numbers given to baseline and system testing in the example assessment template shown in FIG. 2.

[0116] The factors may comprise proportions by which the base time of baseline test and/or system test may be multiplied to reflect the effect of the testing attribute in increasing, decreasing, or not affecting, the average base time of the baseline testing and/or system testing during a migration process. The effect of each testing attribute upon the base time requirement for baseline testing and system testing, and hence the development of each testing time factor, ${}^{i}\text{ъ}_{r}$, will be readily appreciated by those skilled in the art.

[0117] A single universal testing time factor matrix may be used, or many testing time factor matrices may be used, each one containing factors for only one type of testing attribute, such as re-building the application, use of unit testing, creation of testing programs, testing sites, testing criteria and the like. Species of testing attributes may be particularized (such as no re-building, progressive unit testing, perform baseline test off-site and the like). Species of testing attributes may be non-particularized (such as, standard testing suites, non-standard performance criteria, custom hardware simulation during testing, and the like). Particularized and non-particularized species may be combined on a single testing time factor matrix.

[0118] Once at least one testing time factor matrix is identified containing the testing attributes, 1…r, which were received in accordance with step **306**, the testing time factors, ${}^{i}\text{ъ}_1 \ldots {}^{i}\text{ъ}_r$, for baseline and system testing are obtained from the testing time factor matrix or matrices. Note that ${}^{i}\text{ъ}_1 \ldots {}^{i}\text{ъ}_r$, may differ for each type of testing.

[0119] The time requirement for the migration is estimated, in accordance with step **309**,. The time requirement $f(t_i)$ for each individual migration task shown on the assessment template, other than baseline test and system test, is then estimated as follows:

$$f(t_i) = {}^{i}\text{т}\,{}^{i}\psi_{S\text{-}T}\,({}^{i}\text{н}_1\,{}^{i}\text{н}_2 \ldots {}^{i}\text{н}_k)({}^{i}\theta_1\,{}^{i}\theta_2 \ldots {}^{i}\theta_m)\,({}^{i}\text{д}_1\,{}^{i}\text{д}_2 \ldots {}^{i}\text{д}_n)({}^{i}\chi_1\,{}^{i}\chi_2 \ldots {}^{i}\chi_p)$$

[0120] The time requirement for baseline test is estimated as follows:

$$f(t_2) = {}^{2}\text{т}\,{}^{2}\psi_{S\text{-}T}\,({}^{2}\text{н}_1\,{}^{2}\text{н}_2 \ldots {}^{2}\text{н}_k)({}^{2}\theta_1\,{}^{2}\theta_2 \ldots {}^{2}\theta_m)\,({}^{2}\text{д}_1\,{}^{2}\text{д}_2 \ldots {}^{2}\text{д}_n)({}^{2}\chi_1\,{}^{2}\chi_2 \ldots {}^{2}\chi_p)({}^{2}\text{ъ}_1 \ldots {}^{2}\text{ъ}_r)$$

[0121] The time requirement for system test is estimated as follows:

$$f(t_4) = {}^{4}\text{т}\,{}^{4}\psi_{S\text{-}T}\,({}^{4}\text{н}_1\,{}^{4}\text{н}_2 \ldots {}^{4}\text{н}_k)({}^{4}\theta_1\,{}^{4}\theta_2 \ldots {}^{4}\theta_m)\,({}^{4}\text{д}_1\,{}^{4}\text{д}_2 \ldots {}^{4}\text{д}_n)({}^{4}\chi_1\,{}^{4}\chi_2 \ldots {}^{4}\chi_p)({}^{4}\text{ъ}_1 \ldots {}^{4}\text{ъ}_r)$$

[0122] The time estimate for the migration process shown on the assessment template is estimated as follows:

$$\sum_{i=1}^{i} f(t_i)$$

[0123] In accordance with step **310**, desired tolerances may be applied in the manner described with reference to FIG. 1. In accordance with step **311**, a Type C assessment is output (printed or displayed), which shows estimated costs and/or time requirements for the migration process and each migration task in the assessment template, given the base variables and attributes received. Any such estimates may be represented as fixed numbers or ranges. In the preferred embodiment of the invention, costs and/or time requirements estimated in a Type C assessment are output as ranges.

[0124] FIG. 14 is a flow diagram illustrating a method for returning a Type B assessment for the migration of an application from a source platform to a target platform, in accordance with the current invention. In accordance with step **1401**, base variables are received, which comprise an application to be migrated, a source platform on which the application currently operates, and a target platform on which the application is to be re-hosted.

[0125] The application variable may comprise a specific application name and version. Alternatively, the application may include the function of the application and (if more than one function) its component parts, such as production control, batch processing, data mining, online transaction processing (OLTP), or other functions. In the preferred embodiment of the invention, the application variable comprises the name, version and function(s) of the application and its component parts.

[0126] The source and target platform variables may comprise any platforms suitable for operating the application that is to be migrated. These platforms may include, for example, UNIX (generic or variants), OpenVMS, IBM AS/400 or iSeries, HP

300 MPE, IBM Mainframe, and other platforms that are readily known to those skilled in the art.

[0127] In accordance with step **1402**, a user's choice of a Type B assessment is received. In accordance with step **1403**, an identifier for at least one migration task is received, as described with reference to FIG. 1. In accordance with step **1404**, an assessment template may be created in the manner described with reference to FIG. 1. The assessment template for a Type B assessment preferably comprises a representation of migration tasks, which is not as high-level as a Type C assessment template but not as detailed as a Type A assessment template.

[0128] In accordance with step **1405**, base costs and/or times are assigned to each migration task. For a Type B assessment, the base variables are preferably compared with at least one base cost matrix in a database. Each base cost matrix sets forth an average base cost for migration tasks, such as those listed previously. A base cost matrix may conform substantially to that shown in FIG. 4A, wherein migration tasks are listed in one column, and a base cost is shown in the row containing a migration task to which the base cost corresponds. A base cost , $^{i}\beta$, is extracted from the database, for each migration task, 'i', received in step **1403**. Note that each base cost matrix may contain base cost data for more migration tasks than are received in step **1403**.

[0129] Each base cost matrix may set forth average costs for migration tasks, with respect to specific applications. Alternatively, each base cost matrix may set forth average base costs for migration tasks, with respect to the degree of commonality of various applications. For example, base cost matrices may be built for standard applications (for example, word processors and data storage), non-standard (such as compilers and communication software), and custom-made applications. Matrices may be built to reflect additional qualitative categories of applications that have a quantifiably distinguishable effect on the cost of at least one migration task.

[0130] Where time requirements are being estimated, base time requirements are assigned to each migration task. For a Type B assessment, the base variables are preferably compared with at least one base time matrix in a database. Each base cost matrix sets forth an average base cost for migration tasks, such as those listed previously. A base time requirement matrix may conform substantially to that shown in FIG. 9A, wherein migration tasks are listed in one column, and a base time requirement is shown in the row containing a migration task to which the base time requirement corresponds. A base time, ^{i}T, is extracted from the database, for each migration task, 'i', received in step **1403**. Note that each base cost matrix may contain base time data for more migration tasks than are received in step **1403**.

[0131] Each base time requirement matrix may set forth average time requirements for migration tasks, with respect to specific applications. Alternatively, each base time matrix may set forth average base time requirements for migration tasks, with respect to the degree of commonality of various applications. For example, base time matrices may be built for standard applications (for example, word processors and data storage), non-standard (such as compilers and communication software), and custom-made applications. Matrices may be built to reflect additional qualitative categories of applications that have a quantifiably distinguishable effect on the time requirement for at least one migration task.

[0132] In accordance with step **1406**, at least one additional attribute is received. The migration attributes may comprise, for example, hardware attributes, current operating system attributes, attributes of the application to be migrated, environment attributes, or testing attributes, as these are described with reference to FIG. 3, above.

[0133] In addition to the attributes described above, code metrics are also received, in accordance with **1407**. Code metrics may comprise those attributes of the source code of the application being migrated, as those are described with reference to FIG. 1, above. Code metrics are separated into two categories: qualitative and numeric. Qualitative code metrics include such parameters as structural integrity of the application, whether lexical functions are used, the degree to which the application is dependent upon its current operating system, and other code metrics that are not intended to express an actual quantity (even though degrees of a qualitative code metric may be identified by representative numbers). Numeric code metrics include such parameters as code line inventories, code module inventories, file inventories, call and call type inventories, data volume, and other measurable quantities of source code attributes. Code metrics are described further with reference to assignment of factors in step **1408**, below.

[0134] Code metrics may be received directly from a user. Alternatively, the source code may be received and analyzed using a suitable utility program for analyzing application source code and returning metrics such as those described herein. The source code may be received via remote electronic transmission to a computing device – via ftp or e-mail, for example – or physical delivery of computer-readable media followed by entry onto a computer system. In one embodiment, the source code is delivered on computer-readable media and then disposed onto a server computer, desktop computer, laptop computer, or other computing device suitable for assessing the source code. In another embodiment, the code may be transmitted via e-mail or ftp to an assessment server or other computing device suitable for analyzing the source code.

[0135] The metric utility may comprise any computer-implemented utility suitable for analyzing application source code and returning code metrics, such as line counts, per type of code; keyword inventories, whether by individual keyword or categories of keywords, such as procedural keywords, data keywords, environment keywords, identification keywords and the like; inventories of files and directory types, such as source code files, copybook files, COM files and the like; or lexical functions; structural integrity; data volume; operating system dependencies; calls and call types to keywords or lexical functions; and other measurable properties of source code that may impact upon the time and cost for migrating an application from a source platform to a target platform.

[0136] In accordance with step **1408**, cost and/or time factors corresponding to each attribute received in step **1406**, and corresponding to the base variables, are assigned to each migration task received in step **1403**. For a Type B assessment, cost factors corresponding to complexity, operating system attributes, hardware attributes, application attributes, environment attributes, and testing attributes, are assigned using matrices, in the manners described with reference to FIG. 3.

[0137] Additionally, cost factors corresponding to code metrics received in step **1407** are also assigned to the migration tasks received in step **1403**. Qualitative code metrics obtained are compared with at least one code cost factor matrix. Each code cost factor matrix sets forth factors, comprising the amounts by which the costs of various migration tasks are changed due to the presence of at least one qualitative attribute in the source code of the application to be re-hosted. A code cost factor matrix may conform substantially to that shown in FIG. 15, wherein a qualitative code cost factor ${}^{i}\kappa_q$ is shown at the intersection of each row containing a migration task, 'i', and a column containing a qualitative code metric, 'q'. Note that a code cost factor matrix may contain code cost factors for more migration tasks than are received in step **1403**.

[0138] The qualitative code cost factors may comprise proportions by which the base cost of migration tasks may be multiplied to reflect the effect of the qualitative code metric in increasing, decreasing, or not affecting, the average base cost for the migration tasks during the migration process. The effect of each qualitative code metric upon the average base cost for each migration task, and hence the development of each qualitative code cost factor, ${}^{i}\kappa_q$, will be readily appreciated by those skilled in the art.

[0139] A single universal code cost factor matrix may be used, or many code cost factor matrices may be used, each one containing only one type of qualitative code metric. Qualitative code metrics may be expressed in dual form (such as, structural integrity = YES; operating system dependent = NO; lexical functions = YES).

Alternatively, qualitative code metrics may be categorized by degree (such as, low structural integrity; negligible dependence upon operating system; high use of lexical functions). Dual and categorized qualitative code metric representations may be combined on a single code cost factor matrix.

[0140] Once at least one code cost factor matrix is identified containing the qualitative code metrics, 1...q, which were received in accordance with step **1407**, the qualitative code cost factors, ${}^{i}\kappa_{1}$ ... ${}^{i}\kappa_{q}$, for each migration task in the assessment template are obtained from the code cost factor matrix or matrices. Note that ${}^{i}\kappa_{1}$ ... ${}^{i}\kappa_{q}$ may differ for each migration task.

[0141] Because an actual quantity is returned for numeric code metrics, the effect of a numeric code metric, 's', on the cost of a migration task, 'i', may be accounted for formulaically. As the value returned for the metric (such as number of code lines) increases or decreases, then a numeric code cost factor, ${}^{i}\phi_{s}$, may simply be calculated by an appropriate function, or a group of functions that differ for each type of numeric code metric, 1...s. The functions used to calculate each numeric code cost factor, ${}^{i}\phi_{s}$, may have linear elements, geometric elements, differential elements, or any combination of the three. Additionally, the functions may reflect benchmarks, wherein a numeric code cost factor remains constant over a range of values for the numeric code metric. The effect of each numeric code metric upon the average base cost for each migration task, and hence the formulae for calculating each numeric code cost factor, ${}^{i}\phi_{s}$, will be readily appreciated by those skilled in the art.

[0142] Once factors for each numeric code metric, 1...s, which were received in accordance with step **1407**, are calculated for each migration task 'i', then a set of numeric code cost factors, $^{i}\phi_1 \ldots {}^{i}\phi_s$, is obtained for each migration task. Note that the numeric code cost factors $^{i}\phi_1 \ldots {}^{i}\phi_s$ may differ for each migration task.

[0143] In accordance with step **1409**, the estimated cost $f(c_i)$ for each migration task in the assessment template is calculated. The calculation of estimated cost is achieved by applying the cost factors assigned in step **1408** to the base costs assigned in step **1405**. Integrating the factors assigned for the code metrics received in step **1407**, with the equations described with reference to FIG. 3, the cost $f(c_i)$ for each individual migration task received in step **1403**, other than baseline test and system test, is then estimated as follows:

$$f(c_i) = {}^{i}\beta\, {}^{i}\lambda_{S\text{-}T}\, ({}^{i}\hbar_1\, {}^{i}\hbar_2 \ldots {}^{i}\hbar_n)({}^{i}\omega_1\, {}^{i}\omega_2 \ldots {}^{i}\omega_m)({}^{i}\alpha_1\, {}^{i}\alpha_2 \ldots {}^{i}\alpha_n)({}^{i}\epsilon_1\, {}^{i}\epsilon_2 \ldots {}^{i}\epsilon_p)({}^{i}\kappa_1\, {}^{i}\kappa_2 \ldots {}^{i}\kappa_q)({}^{i}\phi_1\, {}^{i}\phi_2 \ldots {}^{i}\phi_s)$$

[0144] The cost $f(c_2)$ for baseline test is estimated as follows:

$$f(c_2) = {}^{2}\beta\, {}^{2}\lambda_{S\text{-}T}\, ({}^{2}\hbar_1\, {}^{2}\hbar_2 \ldots {}^{2}\hbar_n)({}^{2}\omega_1\, {}^{2}\omega_2 \ldots {}^{2}\omega_m)({}^{2}\alpha_1\, {}^{2}\alpha_2 \ldots {}^{2}\alpha_n)({}^{2}\epsilon_1\, {}^{2}\epsilon_2 \ldots {}^{2}\epsilon_p)({}^{2}\kappa_1\, {}^{2}\kappa_2 \ldots {}^{2}\kappa_q)({}^{2}\phi_1\, {}^{2}\phi_2 \ldots {}^{2}\phi_s)\, ({}^{2}\eta_1 \ldots {}^{2}\eta_r).$$

[0145] The cost $f(c_4)$ for system test is estimated as follows:

$$f(c_4) = {}^{4}\beta\, {}^{4}\lambda_{S\text{-}T}\, ({}^{4}\hbar_1\, {}^{4}\hbar_2 \ldots {}^{4}\hbar_n)({}^{4}\omega_1\, {}^{4}\omega_2 \ldots {}^{4}\omega_m)({}^{4}\alpha_1\, {}^{4}\alpha_2 \ldots {}^{4}\alpha_n)({}^{4}\epsilon_1\, {}^{4}\epsilon_2 \ldots {}^{4}\epsilon_p)({}^{4}\kappa_1\, {}^{4}\kappa_2 \ldots {}^{4}\kappa_q)({}^{4}\phi_1\, {}^{4}\phi_2 \ldots {}^{4}\phi_s)({}^{4}\eta_1 \ldots {}^{4}\eta_r).$$

[0146] The cost estimate for the migration process shown on the assessment template is estimated as follows:

$$\sum^{i} f(c_i)$$

i=1

[0147] Where time requirements are estimated, step **1408** includes assignment of time factors to each migration task. For a Type B assessment, time factors corresponding to complexity, operating system attributes, hardware attributes, application attributes, environment attributes, and testing attributes, are assigned using matrices, in the manners described with reference to FIG. 3.

[0148] Additionally, time factors corresponding to code metrics received in step **1407** are also assigned to the migration tasks received in step **1403**. Qualitative code metrics are compared with at least one code time factor matrix. Each code time factor matrix sets forth factors, comprising the amounts by which the time requirements of various migration tasks are changed due to the presence of at least one qualitative attribute in the source code of the application to be re-hosted. A code time factor matrix may conform substantially to that shown in FIG. 17, wherein a qualitative code time factor, ${}^{i}\mu_{q}$, is shown at the intersection of each row containing a migration task, 'i', and a column containing a specie of qualitative code metric, 'q'. Note that a code time factor matrix may contain code time factors for more migration tasks than are received in step **1403**.

[0149] The qualitative code time factors may comprise proportions by which the base time requirements for migration tasks may be multiplied to reflect the effect of the qualitative code metric in increasing, decreasing, or not affecting, the average base time requirement for the migration tasks during the migration process. The effect of each qualitative code metric upon the average base time requirement for each migration task, and hence the development of each qualitative code time factor, ${}^{i}\mu_q$, will be readily appreciated by those skilled in the art.

[0150] A single universal code time factor matrix may be used, or many code time factor matrices may be used, each one containing only one type of qualitative code metric. Qualitative code metrics may be expressed in dual form (such as, structural integrity = YES; operating system dependent = NO; lexical functions = YES). Alternatively, species of qualitative code metrics may be categorized by degree (such as, low structural integrity; negligible dependence upon operating system; high use of lexical functions). Dual and categorized qualitative code metric representations may be combined on a single code time factor matrix.

[0151] Once at least one code time factor matrix is identified containing the qualitative code metrics, 1...q, which were returned in accordance with step **1407**, the qualitative code time factors, ${}^{i}\mu_1 \ldots {}^{i}\mu_q$, for each migration task in the assessment template are obtained from the code time factor matrix or matrices. Note that ${}^{i}\mu_1 \ldots {}^{i}\mu_q$ may differ for each migration task.

[0152] Because an actual quantity is returned for numeric code metrics, the effect of a numeric code metric, 's', on the time requirement for a migration task, 'i', may be accounted for formulaically. As the value returned for the metric (such as number of code lines) increases or decreases, then a numeric code time factor, ${}^{i}\nu_s$, may simply be calculated by an appropriate function, or a group of functions that differ for each type of numeric code metric, 1...s. The functions used to calculate each numeric code time factor, ${}^{i}\nu_s$, may have linear elements, geometric elements, differential elements, or any combination of the three. Additionally, the functions may reflect benchmarks, wherein a numeric code time factor remains constant over a range of values for the numeric code metric. The effect of each numeric code metric upon the average base time requirement for each migration task, and hence the formulae for calculating each numeric code time factor, ${}^{i}\nu_{s}$, will be readily appreciated by those skilled in the art.

[0153] Once factors for each numeric code metric, 1...s, which were returned in accordance with step **1409**, are calculated for each migration task 'i', then a set of numeric code time factors, ${}^{i}\nu_{1} \ldots {}^{i}\nu_{s}$, is obtained for each migration task. Note that ${}^{i}\nu_{1} \ldots {}^{i}\nu_{s}$ may differ for each migration task.

[0154] In accordance with step **1410**, the time requirement for the migration is estimated. The calculation of estimated time is achieved by applying the time factors assigned in step **1408** to the base costs assigned in step **1405**. Integrating the time factors assigned for the code metrics received in step **1407**, with the equations described with reference to FIG. 3, the time requirement $f(t_i)$ for each individual migration task shown on the assessment template, other than baseline test and system test, is estimated as follows:

$$f(t_i) = {}^{i}\text{т}\ {}^{i}\psi_{S\text{-}T}\,({}^{i}\text{н}_1\,{}^{i}\text{н}_2 \ldots {}^{i}\text{н}_k)({}^{i}\theta_1\,{}^{i}\theta_2 \ldots {}^{i}\theta_m)\,({}^{i}\text{д}_1\,{}^{i}\text{д}_2 \ldots {}^{i}\text{д}_n)({}^{i}\chi_1\,{}^{i}\chi_2 \ldots {}^{i}\chi_p)({}^{i}\mu_1\,{}^{i}\mu_{2 \ldots}\,{}^{i}\mu_q)({}^{i}\nu_1\,{}^{i}\nu_{2 \ldots}\,{}^{i}\nu_s)$$

[0155] The time requirement for baseline test is estimated as follows:

$$f(t_2) = {}^{2}\text{т}\ {}^{2}\psi_{S\text{-}T}\,({}^{2}\text{н}_1\,{}^{2}\text{н}_2 \ldots {}^{2}\text{н}_k)({}^{2}\theta_1\,{}^{2}\theta_2 \ldots {}^{2}\theta_m)\,({}^{2}\text{д}_1\,{}^{2}\text{д}_2 \ldots {}^{2}\text{д}_n)({}^{2}\chi_1\,{}^{2}\chi_2 \ldots {}^{2}\chi_p)({}^{2}\mu_1\,{}^{2}\mu_{2 \ldots}\,{}^{2}\mu_q)({}^{2}\nu_1\,{}^{2}\nu_{2 \ldots}\,{}^{2}\nu_s)({}^{2}\text{ъ}_1 \ldots {}^{2}\text{ъ}_r)$$

[0156] The time requirement for system test is estimated as follows:

$$f(t_4) = {}^4\text{т}\, {}^4\psi_{S\text{-}T}\, ({}^4\text{н}_1\, {}^4\text{н}_2 \ldots {}^4\text{н}_k)({}^4\theta_1\, {}^4\theta_2 \ldots\, {}^4\theta_m)\, ({}^4\text{д}_1\, {}^4\text{д}_2 \ldots {}^4\text{д}_n)({}^4\chi_1\, {}^4\chi_2 \ldots\, {}^4\chi_p)({}^4\mu_1\, {}^4\mu_{2\ldots}\, {}^4\mu_q)({}^4\nu_1\, {}^4\nu_{2\ldots}\, {}^4\nu_s)({}^4\text{ъ}_1 \ldots {}^4\text{ъ}_r)$$

[0157] The time estimate for the migration process shown on the assessment template is estimated as follows:

$$\sum_{i=1}^{i} f(t_i)$$

[0158] In accordance with step **1411**, desired tolerances are applied in the manner described with reference to FIG. 1. In accordance with step **1412**, a Type B assessment is output (printed or displayed), which shows estimated costs and/or time requirements for the migration process and each migration task in the assessment template, given the base variables, attributes and code metrics received. Any such estimates may be represented as fixed numbers or ranges. In the preferred embodiment of the invention, costs and/or time requirements estimated in a Type B assessment are output as ranges.

[0159] FIG. 16 is a flow diagram illustrating a method for returning a Type A assessment for the migration of an application from a source platform to a target platform, in accordance with the current invention. In accordance with step **1601**, base variables are received, which comprise an application to be migrated, a source platform on which the application currently operates, and a target platform on which the application is to be re-hosted.

[0160] The application variable may comprise a specific application name and version. Alternatively, the application may include the function of the application and (if more than one function) its component parts, such as production control, batch processing, data mining, online transaction processing (OLTP), or other functions. In the preferred embodiment of the invention, the application variable comprises the name, version and function(s) of the application and its component parts.

[0161] The source and target platform variables may comprise any platforms suitable for operating the computer-based application to be migrated. These platforms may include, for example, UNIX (generic or variants), OpenVMS, IBM AS/400 or iSeries, HP 300 MPE, IBM Mainframe, and other platforms that are readily known to those skilled in the art.

[0162] In accordance with step **1602**, a user's choice of a Type B assessment is received. In accordance with step **1603**, an identifier for at least one migration task is received, as described with reference to FIG. 1. In accordance with step **1604**, an assessment template may be created in the manner described with reference to FIG. 1. The assessment template should contain both high-level tasks and the low-level tasks associated with each, such that cost and time requirements will be estimated in great detail, when compared to a Type C or Type B assessment.

[0163] In accordance with step **1605**, base costs and/or times are assigned to each migration task. For a Type A assessment, the base variables are preferably compared with at least one base cost matrix in a database. Each base cost matrix sets forth an average base cost for migration tasks, such as those listed previously. A base cost matrix may conform substantially to that shown in FIG. 4A, wherein migration tasks are listed in one column, and a base cost is shown in the row containing a migration task to which the base cost corresponds. A base cost , $^{i}\beta$, is extracted from the database, for each migration task, 'i', received in step **1603**. Note that each base cost matrix may contain base cost data for more migration tasks than are received in step **1603**.

[0164] Each base cost matrix may set forth average costs for migration tasks, with respect to specific applications. Alternatively, each base cost matrix may set forth average base costs for migration tasks, with respect to the degree of commonality of various applications. For example, base cost matrices may be built for standard applications (for example, word processors and data storage), non-standard (such as compilers and communication software), and custom-made applications. Matrices may be built to reflect additional qualitative categories of applications that have a quantifiably distinguishable effect on the cost of at least one migration task.

[0165] Where time requirements are being estimated, base time requirements are assigned to each migration task. For a Type A assessment, the base variables are preferably compared with at least one base time matrix in a database. Each base cost matrix sets forth an average base cost for migration tasks, such as those listed previously. A base time requirement matrix may conform substantially to that shown in FIG. 9A, wherein migration tasks are listed in one column, and a base time requirement is shown in the row containing a migration task to which the base time requirement corresponds. A base time, $^{i}\tau$, is extracted from the database, for each migration task, 'i', received in step **1603**. Note that each base cost matrix may contain base time data for more migration tasks than are received in step **1603**.

[0166] Each base time requirement matrix may set forth average time requirements for migration tasks, with respect to specific applications. Alternatively, each base time matrix may set forth average base time requirements for migration tasks, with respect to the degree of commonality of various applications. For example, base time matrices may be built for standard applications (for example, word processors and data storage), non-standard (such as compilers and communication software), and custom-made applications. Matrices may be built to reflect additional qualitative categories of applications that have a quantifiably distinguishable effect on the time requirement for at least one migration task.

[0167] In accordance with step **1606**, at least one migration attribute is received. The migration attributes may comprise, for example, hardware attributes, current operating system attributes, attributes of the application to be migrated, environment attributes, or testing attributes, as these are described with reference to FIG. 3, above. To achieve the level of detail required for a Type A assessment to be distinguishable from a Type B assessment, however, additional attributes may need to be considered.

[0168] For a Type A assessment, additional hardware attributes are received that will enable a more accurate cost analysis and plan. Such additional hardware attributes may include, for example, whether the application operates on multiple systems; user connection types (dial-up, serial, WAN, LAN, etc.); and the existence and currency of hardware architecture documentation.

[0169] For a Type A assessment, additional application attributes are received that will enable a more accurate cost analysis and plan. Such additional application attributes may include, for example, unique porting complexities; and whether previous unsuccessful portings have been attempted. Such additional application attributes may also include the existence and currency of application documentation, such as documentation for system and functional requirements; application architecture diagrams; subsystem top-level diagrams; module hierarchical diagrams; user documentation; Y2K documentation; and programming standards documentation.

[0170] For a Type A assessment, additional environment attributes are received that will enable a more accurate cost analysis and plan. Such additional environment attributes may include, for example, how the application is partitioned (client/server, multi-threaded, one large module, discrete modules executable via CALL or ECTL); embedded SQL or DL/I access; and whether the application is X/Open XA compliant.

[0171] Such additional environment attributes may also include, for example, configuration management tools on the source platform (e.g., Panvalet, Librarian); preferred configuration management tools on the target platform; the last re-building of the application; executable images in the application; and whether the application links with non-standard object libraries.

[0172] Such additional environment attributes may also include operating system dependencies, such as multinational character sets; user-written ISPF panels; inter-process communication; SYSPLEX functionality; MACRO calls; CSP map usage; internal EBCDIC coding dependencies; and other such dependencies.

[0173] Such additional environment attributes may also include data storage and access attributes other than type, size and configuration of databases and tables, such as needs to access archived data after migration; timing constraints for cutover; whether data can be accessed on media other than disks; import/export data from/to outside systems; type and size of native files; whether native language I/O are used to access native data files; and whether the application shares data sets with other systems.

[0174] Such additional environment attributes may also include third party products used to implement or augment any of the environment attributes described herein.

[0175] In accordance with step **1607**, the source code of the application to be migrated is received. The source code may be received via remote electronic transmission to a computing device – via ftp or e-mail, for example – or physical delivery of computer-readable media followed by entry onto a computer system. In one embodiment, the source code is delivered on computer-readable media and then disposed onto a server computer, desktop computer, laptop computer, or other computing device suitable for assessing the source code. In another embodiment, the code may be transmitted via e-mail or ftp to an assessment server or other computing device suitable for analyzing the source code.

[0176] In accordance with step **1608**, metrics are returned for the source code received in step **1607**. This may be accomplished by analyzing the source code using a metric utility. The code metrics to be returned may be defined prior to or after analysis of the source code, and may include qualitative and/or numeric code metrics, such as those described previously. The metric utility may comprise any computer-implemented utility suitable for analyzing application source code and returning code metrics, such as line counts, per type of code; keyword inventories, whether by individual keyword or categories of keywords, such as procedural keywords, data keywords, environment keywords, identification keywords and the like; inventories of files and directory types, such as source code files, copybook files, COM files and the like; or lexical functions; structural integrity; data volume; operating system dependencies; calls and call types to keywords or lexical functions; and other measurable properties of source code that may impact upon the time and cost for migrating an application from a source platform to a target platform.

[0177] In accordance with step **1609**, cost and/or time factors corresponding to each attribute received in step **1606**, and corresponding to the base variables, are assigned to each migration task received in step **1603**. For a Type A assessment, cost factors corresponding to complexity, operating system attributes, hardware attributes, application attributes, environment attributes, and testing attributes, are assigned using matrices, in the manners described with reference to FIG. 3.

[0178] Additionally, cost factors corresponding to code metrics received in step **1608** are also assigned to the migration tasks received in step **1603**. Qualitative code metrics obtained are compared with at least one code cost factor matrix. Each code cost factor matrix sets forth factors, comprising the amounts by which the costs of various migration tasks are changed due to the presence of at least one qualitative attribute in the source code of the application to be re-hosted. A code cost factor matrix may conform substantially to that shown in FIG. 15, wherein a qualitative code cost factor ${}^{i}\kappa_q$ is shown at the intersection of each row containing a migration task, 'i', and a column containing a qualitative code metric, 'q'. Note that a code cost factor matrix may contain code cost factors for more migration tasks than are received in step **1603**.

[0179] The qualitative code cost factors may comprise proportions by which the base cost of migration tasks may be multiplied to reflect the effect of the qualitative code metric in increasing, decreasing, or not affecting, the average base cost for the migration tasks during the migration process. The effect of each qualitative code metric upon the average base cost for each migration task, and hence the development of each qualitative code cost factor, ${}^{i}\kappa_q$, will be readily appreciated by those skilled in the art.

[0180] A single universal code cost factor matrix may be used, or many code cost factor matrices may be used, each one containing only one type of qualitative code metric. Qualitative code metrics may be expressed in dual form (such as, structural integrity = YES; operating system dependent = NO; lexical functions = YES). Alternatively, qualitative code metrics may be categorized by degree (such as, low structural integrity; negligible dependence upon operating system; high use of lexical functions). Dual and categorized qualitative code metric representations may be combined on a single code cost factor matrix.

[0181] Once at least one code cost factor matrix is identified containing the qualitative code metrics, 1...q, which were received in accordance with step **1608**, the qualitative code cost factors, ${}^{i}\kappa_1 \ldots {}^{i}\kappa_q$, for each migration task in the assessment template are obtained from the code cost factor matrix or matrices. Note that ${}^{i}\kappa_1 \ldots {}^{i}\kappa_q$ may differ for each migration task.

[0182] Because an actual quantity is returned for numeric code metrics, the effect of a numeric code metric, 's', on the cost of a migration task, 'i', may be accounted for formulaically. As the value returned for the metric (such as number of code lines) increases or decreases, then a numeric code cost factor, ${}^{i}\phi_s$, may simply be calculated by an appropriate function, or a group of functions that differ for each type of numeric code metric, 1...s. The functions used to calculate each numeric code cost factor, ${}^{i}\phi_s$, may have linear elements, geometric elements, differential elements, or any combination of the three. Additionally, the functions may reflect benchmarks, wherein a numeric code cost factor remains constant over a range of values for the numeric code metric. The effect of each numeric code metric upon the average base cost for each migration task, and hence the formulae for calculating each numeric code cost factor, ${}^{i}\phi_s$, will be readily appreciated by those skilled in the art.

[0183] Once factors for each numeric code metric, 1...s, which were received in accordance with step **1608**, are calculated for each migration task 'i', then a set of numeric code cost factors, ${}^{i}\phi_1 \ldots {}^{i}\phi_s$, is obtained for each migration task. Note that the numeric code cost factors ${}^{i}\phi_1 \ldots {}^{i}\phi_s$ may differ for each migration task.

[0184] In accordance with step **1610**, a plurality of customized cost factors ${}^{i}\gamma_1 \ldots {}^{i}\gamma_t$ are received from a user for at least one migration task, 'i', and at least one additional attribute, 't'. Customized cost factors may be received from a user, in relation to the attributes described with reference to step **1606**, or the code metrics returned in step **1608**, in place of any factors obtained using the matrices or calculations described herein. The customized cost factors may also relate to attributes not contained in such matrices, which the user desires to include in cost estimation. The customized cost factors, ${}^{i}\gamma_1 \ldots {}^{i}\gamma_t$, may be received in list form, or they may be filled in on a matrix template or multiple matrix templates, for storage and later reference.

[0185] Customized cost factors may comprise proportions by which the base cost of migration tasks may be multiplied to reflect the effect of additional attributes in increasing, decreasing, or not affecting, the average base cost of the migration tasks during a migration process. The effect of each additional attribute upon the base cost of each migration task, and hence the development of each customized cost factor, ${}^{i}\gamma_t$, will be readily appreciated by those skilled in the art. Note that ${}^{i}\gamma_1 \ldots {}^{i}\gamma_t$ may differ for each migration task.

[0186] In accordance with step **1611**, the estimated cost $f(c_i)$ for each migration task in the assessment template is calculated. The calculation of estimated cost is achieved by applying the cost factors assigned in step **1609** and the custom factors received in **1610** to the base costs assigned in step **1605**. Integrating the factors assigned for the code metrics received in step **1609**, and the custom cost factors received in step **1610**, with the equations described with reference to FIG. 3, the cost $f(c_i)$ for each individual migration task received in step **1603**, other than baseline test and system test, is estimated as follows:

$$f(c_i) = {}^{i}\beta\, {}^{i}\lambda_{S\text{-}T}\, ({}^{i}\hbar_1\, {}^{i}\hbar_2 \ldots {}^{i}\hbar_n)({}^{i}\omega_1\, {}^{i}\omega_2 \ldots {}^{i}\omega_m)({}^{i}\alpha_1\, {}^{i}\alpha_2 \ldots {}^{i}\alpha_n)({}^{i}\epsilon_1\, {}^{i}\epsilon_2 \ldots {}^{i}\epsilon_p)\, ({}^{i}\kappa_1\, {}^{i}\kappa_2 \ldots {}^{i}\kappa_q)({}^{i}\phi_1\, {}^{i}\phi_2 \ldots {}^{i}\phi_s)({}^{i}\gamma_1\, {}^{i}\gamma_2 \ldots {}^{i}\gamma_t)$$

[0187] The cost $f(c_2)$ for baseline test is estimated as follows:

$$f(c_2) = {}^2\beta\, {}^2\lambda_{S\text{-}T}\, ({}^2\hbar_1\, {}^2\hbar_2 \ldots {}^2\hbar_n)({}^2\omega_1\, {}^2\omega_2 \ldots {}^2\omega_m)({}^2\alpha_1\, {}^2\alpha_2 \ldots {}^2\alpha_n)({}^2\epsilon_1\, {}^2\epsilon_2 \ldots {}^2\epsilon_p)({}^2\kappa_1\, {}^2\kappa_2 \ldots {}^2\kappa_q)({}^2\phi_1\, {}^2\phi_2 \ldots {}^2\phi_s)({}^2\gamma_1\, {}^2\gamma_2 \ldots {}^i\gamma_t)\, ({}^2\eta_1 \ldots {}^2\eta_r).$$

[0188] The cost $f(c_4)$ for system test is estimated as follows:

$$f(c_4) = {}^4\beta\, {}^4\lambda_{S\text{-}T}\, ({}^4\hbar_1\, {}^4\hbar_2 \ldots {}^4\hbar_n)({}^4\omega_1\, {}^4\omega_2 \ldots {}^4\omega_m)({}^4\alpha_1\, {}^4\alpha_2 \ldots {}^4\alpha_n)({}^4\epsilon_1\, {}^4\epsilon_2 \ldots {}^4\epsilon_p)({}^4\kappa_1\, {}^4\kappa_2 \ldots {}^4\kappa_q)({}^4\phi_1\, {}^4\phi_2 \ldots {}^4\phi_s)\, ({}^4\gamma_1\, {}^4\gamma_2 \ldots {}^4\gamma_t)({}^4\eta_1 \ldots {}^4\eta_r).$$

[0189] The cost estimate for the migration process shown on the assessment template is estimated as follows:

$$\sum_{i=1}^{i} f(c_i)$$

[0190] Where time requirements are estimated, step **1609** includes assignment of time factors to each migration task. For a Type A assessment, time factors corresponding to complexity, operating system attributes, hardware attributes, application attributes, environment attributes, and testing attributes, are assigned using matrices, in the manners described with reference to FIG. 3.

[0191] Additionally, time factors corresponding to code metrics returned in step **1608** are also assigned to the migration tasks received in step **1603**. Qualitative code metrics are compared with at least one code time factor matrix. Each code time factor matrix sets forth factors, comprising the amounts by which the time requirements of various migration tasks are changed due to the presence of at least one qualitative attribute in the source code of the application to be re-hosted. A code time factor matrix may conform substantially to that shown in FIG. 17, wherein a qualitative code time factor, ${}^i\mu_q$, is shown at the intersection of each row containing a migration task, 'i', and a column containing a specie of qualitative code metric, ‘q’. Note that a code time factor matrix may contain code time factors for more migration tasks than are received in step **1603**.

[0192] The qualitative code time factors may comprise proportions by which the base time requirements for migration tasks may be multiplied to reflect the effect of the qualitative code metric in increasing, decreasing, or not affecting, the average base time requirement for the migration tasks during the migration process. The effect of each qualitative code metric upon the average base time requirement for each migration task, and hence the development of each qualitative code time factor, ${}^{i}\mu_{q}$, will be readily appreciated by those skilled in the art.

[0193] A single universal code time factor matrix may be used, or many code time factor matrices may be used, each one containing only one type of qualitative code metric. Qualitative code metrics may be expressed in dual form (such as, structural integrity = YES; operating system dependent = NO; lexical functions = YES). Alternatively, species of qualitative code metrics may be categorized by degree (such as, low structural integrity; negligible dependence upon operating system; high use of lexical functions). Dual and categorized qualitative code metric representations may be combined on a single code time factor matrix.

[0194] Once at least one code time factor matrix is identified containing the qualitative code metrics, 1…q, which were returned in accordance with step **1608**, the qualitative code time factors, ${}^{i}\mu_{1} \ldots {}^{i}\mu_{q}$, for each migration task in the assessment template are obtained from the code time factor matrix or matrices. Note that ${}^{i}\mu_1 \ldots {}^{i}\mu_q$ may differ for each migration task.

[0195] Because an actual quantity is returned for numeric code metrics, the effect of a numeric code metric, 's', on the time requirement for a migration task, 'i', may be accounted for formulaically. As the value returned for the metric (such as number of code lines) increases or decreases, then a numeric code time factor, ${}^{i}\nu_s$, may simply be calculated by an appropriate function, or a group of functions that differ for each type of numeric code metric, 1...s. The functions used to calculate each numeric code time factor, ${}^{i}\nu_s$, may have linear elements, geometric elements, differential elements, or any combination of the three. Additionally, the functions may reflect benchmarks, wherein a numeric code time factor remains constant over a range of values for the numeric code metric. The effect of each numeric code metric upon the average base time requirement for each migration task, and hence the formulae for calculating each numeric code time factor, ${}^{i}\nu_s$, will be readily appreciated by those skilled in the art.

[0196] Once factors for each numeric code metric, 1...s, which were returned in accordance with step **1608**, are calculated for each migration task 'i', then a set of numeric code time factors, ${}^{i}\nu_1 \ldots {}^{i}\nu_s$, is obtained for each migration task. Note that ${}^{i}\nu_1 \ldots {}^{i}\nu_s$ may differ for each migration task.

[0197] Also, where time requirements are estimated, a plurality of customized time factors, ${}^{i}\sigma_1 \ldots {}^{i}\sigma_t$, may be received from a user, in accordance with step **1610**, for at least one migration task, 'i', and at least one additional attribute, 't'. Customized time factors may be received from a user, in relation to the attributes described with reference to step **1606**, or the code metrics returned in step **1608**, in place of any factors obtained using the matrices or calculations described herein. The customized time factors may also relate to attributes not contained in such matrices, which the user desires to include in estimation of time requirements. The customized time factors, ${}^{i}\sigma_{1}\ldots{}^{i}\sigma_{t}$, may be received in list form, or they may be filled in on a matrix template or multiple matrix templates, for storage and later reference.

[0198] Customized time factors may comprise proportions by which the base time of migration tasks may be multiplied to reflect the effect of the additional attribute in increasing, decreasing, or not affecting, the average base time requirement for the migration tasks during a migration process. The effect of each additional attribute upon the base time requirement for each migration task, and hence the development of each customized time factor, ${}^{i}\sigma_{t}$, will be readily appreciated by those skilled in the art. Note that ${}^{i}\sigma_{1}\ldots{}^{i}\sigma_{t}$ may differ for each migration task.

[0199] In accordance with step **1612**, the time requirement for the migration is estimated. The calculation of estimated time is achieved by applying the time factors assigned in step **1609** and the custom factors received in step **1610** to the base costs assigned in step **1605**. Integrating the time factors assigned for the code metrics returned in step **1608**, and the custom time factors received in step **1610**, with the equations described with reference to FIG. 3, the time requirement $f(t_i)$ for each individual migration task shown on the assessment template, other than baseline test and system test, is estimated as follows:

$$f(t_i) = {}^{i}\text{T}\ {}^{i}\psi_{S\text{-}T}({}^{i}\text{H}_1\ {}^{i}\text{H}_2 \ldots {}^{i}\text{H}_k)({}^{i}\theta_1\ {}^{i}\theta_2 \ldots\ {}^{i}\theta_m)\ ({}^{i}\text{Д}_1\ {}^{i}\text{Д}_2 \ldots {}^{i}\text{Д}_n)({}^{i}\chi_1\ {}^{i}\chi_2 \ldots {}^{i}\chi_p)\ ({}^{i}\mu_1\ {}^{i}\mu_{2\ldots}\ {}^{i}\mu_q)({}^{i}\nu_1\ {}^{i}\nu_{2\ldots}\ {}^{i}\nu_s)({}^{i}\sigma_1\ {}^{i}\sigma_{2\ldots}\ {}^{i}\sigma_t)$$

[0200] The time requirement for baseline test is estimated as follows:

$$f(t_2) = {}^{2}\text{T}\ {}^{2}\psi_{S\text{-}T}\ ({}^{2}\text{H}_1\ {}^{2}\text{H}_2 \ldots {}^{2}\text{H}_k)({}^{2}\theta_1\ {}^{2}\theta_2 \ldots {}^{2}\theta_m)({}^{2}\text{Д}_1\ {}^{2}\text{Д}_2 \ldots {}^{2}\text{Д}_n)({}^{2}\chi_1\ {}^{2}\chi_2 \ldots {}^{2}\chi_p)({}^{2}\mu_1\ {}^{2}\mu_{2\ldots}\ {}^{2}\mu_q)({}^{2}\nu_1\ {}^{2}\nu_{2\ldots}\ {}^{2}\nu_s)({}^{2}\sigma_1\ {}^{2}\sigma_{2\ldots}\ {}^{2}\sigma_t)({}^{2}\text{Ъ}_1 \ldots {}^{2}\text{Ъ}_r)$$

[0201] The time requirement for system test is estimated as follows:

$$f(t_4) = {}^{4}\text{T}\ {}^{4}\psi_{S\text{-}T}\ ({}^{4}\text{H}_1\ {}^{4}\text{H}_2 \ldots {}^{4}\text{H}_k)({}^{4}\theta_1\ {}^{4}\theta_2 \ldots {}^{4}\theta_m)({}^{4}\text{Д}_1\ {}^{4}\text{Д}_2 \ldots {}^{4}\text{Д}_n)({}^{4}\chi_1\ {}^{4}\chi_2 \ldots {}^{4}\chi_p)({}^{4}\mu_1\ {}^{4}\mu_{2\ldots}\ {}^{4}\mu_q)({}^{4}\nu_1\ {}^{4}\nu_{2\ldots}\ {}^{4}\nu_s)({}^{4}\sigma_1\ {}^{4}\sigma_{2\ldots}\ {}^{4}\sigma_t)({}^{4}\text{Ъ}_1 \ldots {}^{4}\text{Ъ}_r)$$

[0202] The time estimate for the migration process shown on the assessment template is estimated as follows:

$$\sum_{i=1}^{i} f(t_i)$$

[0203] In accordance with step **1613**, desired tolerances are applied in the manner described with reference to FIG. 1. In accordance with step **1614**, a Type A assessment is output, which shows estimated costs and/or time requirements for the migration process and each migration task in the assessment template, given the base variables, attributes and code metrics received. Any such estimates may be represented as fixed numbers or ranges. In the preferred embodiment of the invention, costs and/or time requirements estimated in a Type A assessment are output as ranges, which are accurate within thirty percent (30%). More preferably, the estimates are accurate within fifteen to twenty-five percent (15-25%).

[0204] The invention is also directed to computer-readable program products having computer-readable program code means for producing cost and time requirement estimates for migration projects, in accordance with the various embodiments of the method described herein. Selecting appropriate media and implementing the process described herein on such media are matters that will be readily known to those skilled in the art.

[0205] While the foregoing material describes and illustrates the current invention with reference to particular embodiments, these embodiments are intended as examples and not to define the entire scope of the current invention. Those skilled in the art will appreciate that many aspects of these embodiments may be changed and steps of the various methods re-arranged, such as creation of template, assignment of base costs and receiving additional data, without departing from the spirit or scope of the invention.